CLAIMS

What is claimed is:

1. A method for managing session information in a client-server environment comprising:

(a) establishing a communication session between a client and a server;

(b) storing session information associated with the communication session between the client and the server in a first log file stored in a persistent volatile memory;

(c) storing the session information in a cache file stored in a volatile memory of the server; and

(d) reconstructing the cache file after a server failure using the session information stored in the first log.

2. The method of claim 1 further comprising the step of identifying the communication session with a session identifier.

3. The method of claim 1 further comprising the step of scanning the session information for statistical purposes.

4. The method of claim 1 further comprising swapping the first log file with a second log file to store the session information when a certain predetermined criteria is met.

5. The method of claim 4 wherein the predetermined criteria further comprises when the size of the first log file exceeds a predefined size.

6. The method of claim 4 wherein the predetermined criteria further comprises when a predetermined amount of time has elapsed.

7. The method of claim 4 further comprising the step of resetting the first log file to an unallocated state.

8. The method of claim 1 further comprising transferring the session information from the cache file stored in the volatile memory to a database located in a persistent mass storage.

9. The method of claim 1 further comprising the steps of:

(e) serializing multiple requests from a client; and

(f) updating the session information for each request.

10. The method of claim 9 wherein step (f) further comprises the step of:

(f-a) creating the session information for each request.

11. The method of claim 1 wherein step (d) further comprises the steps of:

(d-a) locking the server;

(d-b) transferring the session information stored in the first log file to the cache file;

(d-c) transferring the session information in the cache file to a database stored in the persistent volatile memory;

(d-d) deleting the first log file; and

(d-e) scanning the database to recreate information stored in a data structure, wherein the information stored in the data structure comprises an index of the cache file.

12. The method of claim 1 further comprising the step of locking the server.

13. A session storage manager for managing session information in a client-server environment, the session storage manager comprising:

(a) a persistent volatile memory;

(b) a first log file, stored in the persistent volatile memory, containing session information;

(c) a record cache storing a record of session information;

(d) an execution thread appending the session information stored in the record cache to the first log file;

(e) a database cache storing the session information after the session information has been stored in the first log file;

wherein the database cache is reconstructed after a server failure using the session information stored in the first log file.

14. The session storage manager of claim 13 further comprising:

(f) a volatile memory;

(g) a persistent mass storage;

(h) a database having an address space in the persistent mass storage and in communication with the execution thread; and

(i) a flushing thread transferring the session information stored in the database cache to the database.

15. The session storage manager of claim 14 wherein the record cache is stored in the volatile memory.

16. The session storage manager of claim 14 wherein the record cache prevents partial writes of the session information to the database by the flushing thread.

17. The session storage manager of claim 14 wherein the record cache further comprises at least one of a record length, a start magic number, a data length, a database offset, data, and an end magic number.

18. The session storage manager of claim 17 wherein the start magic number and the end magic number are used to verify the validity of the contents of at least one of the record length, the data length, the database offset, and the data.

19. The session storage manager of claim 14 wherein the database further comprises at least one of an allocated block containing active session information, an unallocated block available to store the session information, free memory space, and an offset to the free memory space available to store the session information.

20. The session storage manager of claim 14 wherein the persistent mass storage further comprises the persistent volatile memory.

21. A session storage manager for managing session information in a client-server environment, the session storage manager comprising:

(a) means for establishing a communication session between a client and a server;

(b) means for storing session information for the communication session to a first log file stored in a persistent volatile memory;

(c) means for storing the session information in a cache file stored in a volatile memory; and

(d) means for reconstructing the cache file using the session information stored in the first log.